Queue cheatsheet for coding interviews

Introduction

A queue is a linear collection of elements that are maintained in a sequence and can be modified by the addition of elements at one end of the sequence (enqueue operation) and the removal of elements from the other end (dequeue operation). Usually, the end of the sequence at which elements are added is called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front of the queue. As an abstract data type, queues can be implemented using arrays or singly linked lists.
This behavior is commonly called FIFO (first in, first out). The name "queue" for this type of structure comes from the analogy to people lining up in real life to wait for goods or services.
Breadth-first search is commonly implemented using queues.

Learning resources

- Readings
    - [To Queue Or Not To Queue](), basecs
- Videos
    - [Queues](), University of California San Diego

Implementations

Language	API
C++	`std::queue`
Java	`java.util.Queue`.Use `java.util.ArrayDeque`
Python	`queue`
JavaScrip	N/A

t

Time complexity

Operation	Big-O
Enqueue/Offer	O(1)
Dequeue/Poll	O(1)
Front	O(1)
Back	O(1)
isEmpty	O(1)

Things to look out for during interviews

Most languages don't have a built-in Queue class which can be used, and candidates often use arrays (JavaScript) or lists (Python) as a queue. However, note that the enqueue operation in such a scenario will be O(n) because it requires shifting of all other elements by one. In such cases, you can flag this to the interviewer and say that you assume that there's a queue data structure to use which has an efficient enqueue operation.

Corner cases

- Empty queue
- Queue with one item
- Queue with two items

Essential questions

These are essential questions to practice if you're studying for this topic.
- [Implement Stack using Queues](#)

Recommended practice questions

These are recommended questions to practice after you have studied for the topic and have practiced the essential questions.

- [Implement Queue using Stacks](#)
- [Design Circular Queue](#)
- [Design Hit Counter (LeetCode Premium)](#)